

Бусы

1.5 с / 30 с

1 GB

Аня и Яна сёстры. На Новый Год каждая из них получила в подарок ниточку с разноцветными бусинками. Каждый цвет можно обозначить буквой английского алфавита (“a”...“z”), а каждой ниточке, таким образом, можно сопоставить слово.

Девочки хотят смастерить из своих ниточек бусы. Они могут сделать бусы из любой ниточки, убрав некоторое количество (возможно ноль) бусинок с её концов и затем соединив концы вместе. Получившиеся бусы можно затем поворачивать и переворачивать.

Девочки хотят, чтобы их бусы выглядели совершенно одинаково, и чтобы они были как можно более длинными. Какой наибольшей длины бусы они могут сделать?

Входные данные. И первая и вторая строка содержат по непустой последовательности, состоящей из не более чем N строчных букв, – описания ниточек Ани и Яны, соответственно.

Выходные данные. Первая строка должна содержать одно целое положительное число: максимальное число бусинок в бусах каждой из девочек. Гарантируется, что существует решение с положительной длиной бус.

Вторая строка должна содержать два целых числа: стартовые позиции бус на ниточках Ани и Яны, соответственно. Если вариантов несколько, вывести любой из них. Позиции пронумерованы слева направо, начиная с 0.

Пример. Входные данные	Выходные данные
zxyabcd	4
yxbadctz	3 2

Можно сделать так:

“zxyabcd” → “---abcd”

“yxbadctz” → “--badc--”

Ниточки “abcd” и “badc” порождают идентичные бусы.

Оценивание. В этом задании программа получит полные очки за группу тестов, только если она правильно найдёт самые длинные возможные бусы во всех тестах группы. Если в каждом тесте она найдёт бусы длиной по крайней мере в половину от длиннейших возможных, то она получит 20% от очков.

В группах тестов выполняются следующие условия:

1. (25 очков) $N = 100$.
2. (20 очков) $N = 400$.
3. (40 очков) $N = 3000$.
4. (15 очков) $N = 3000$.

Последняя группа тестов особенная. В ней такой же временной предел, что и выше, но решение может использовать только 3 МВ памяти. По техническим причинам на сервере это подзадание реализовано в виде отдельной задачи (**necklace4**), и решения нужно отправлять отдельно как в **necklace1**, так и в **necklace4**.

Для решений на C и C++ предел в 3 МВ применяется напрямую. Для решений на Java и Python, предел выбирается как память, на 3 МВ превышающая потребности программы “Hello world”. Для Java, вдобавок, сборщик мусора JVM конфигурируется параметрами

`-Xmx4224k -Xss256k -XX:MaxMetaspaceSize=8704k.`